



TITLE:

最小残差法による前処理を用いた GMRES法について (微分方程式 の数値解法と線形計算)

AUTHOR(S):

田中, 祐; 野寺, 隆

CITATION:

田中, 祐 ...[et al]. 最小残差法による前処理を用いたGMRES法について (微分方程式の数値解法と線形計算). 数理解析研究所講究録 2003, 1320: 190-200

ISSUE DATE:

2003-05

URL:

<http://hdl.handle.net/2433/43092>

RIGHT:

最小残差法による前処理を用いた GMRES(m) 法について

慶應義塾大学・理工学部 田中 祐 (Tasuku TANAKA) 野寺 隆 (Takashi NODERA)

Faculty of Science and Technology,
Keio University

1 はじめに

偏微分方程式の境界値問題の近似解を求めることは、熱伝導や流体力学など理工学における様々な分野において現れる問題であり、一般に数値的に解く手法がしばしば用いられる。通常、偏微分方程式の境界値問題を有限要素法や有限差分法などによって離散化し、得られた連立1次方程式

$$Ax = b \quad (1)$$

を解くことになる。ただし、係数行列 A は大型で疎な $n \times n$ の正則な行列である。また、式 (1) に GMRES 法を直接適用すると、近似解が得られるまでに多くの計算時間および反復回数を必要とする場合がある。そこで、行列の前処理を併用することが多い。これは解 x を変えずに係数行列を変換する方法であり、元の方程式 (1) に直接 GMRES(m) 法を適用するよりも、少ない計算時間と反復回数で近似解が求められるからである。行列の前処理には様々な方法があるが、本稿では最小残差法 (minimal residual method, 通常 MR 法と呼ばれている) を用いて係数行列 A の近似逆行列を構成する方法について述べる。

2 最小残差法による前処理

式 (1) に直接反復法を適用しても、収束するまでに多くの計算時間を要したり、場合によっては全く収束しないことがある。これは、一般に反復法の収束性が係数行列の固有値分布と密接に関係していて、この固有値分布によってアルゴリズムの収束性を左右する要因の一つであるからである。そこで式 (1) を

$$MAx = Mb \quad (2)$$

あるいは

$$\begin{cases} AMy = b \\ x = My \end{cases} \quad (3)$$

というように、行列 M による変換を行ってから反復法を適用することがある。ただし、 $M \approx A^{-1}$ とする。この行列 M を前処理行列と呼び、適切な行列 M を選ぶことによって、よ

り少ない計算時間と反復回数で解を求められるからである。通常、式(2)の変換を左側前処理、式(3)の変換を右側前処理と呼ぶ。近似逆行列による前処理では左側前処理、右側前処理のいずれも可能である。しかし、右側前処理は式(3)から明らかなように、元の方程式(1)の残差ベクトルの値を変えないという理由から、本稿では右側前処理を扱うことにする。

この試みは古くは1950年代から始まっており、1970年初頭から1980年代にかけて前処理の研究が大きな成果をあげた。現在では対角行列スケーリング、行列分離、不完全行列分解、多項式による前処理、近似逆行列などが用いられている。近年、このような前処理の中で近似逆行列を利用する研究が活発に行なわれている。本稿では、前処理として利用する近似逆行列を反復法の一つの最小残差法を利用して計算する技法について述べる²⁾。

2.1 近似逆行列による前処理

本節では、残差行列 $R = I - AM$ のフロベニウスノルムを最小にすることに基づく近似逆行列の計算について述べる。まず、次のような目的関数 $F(M)$ を考える^{3, 4)}。

$$F(M) = \|I - AM\|_F^2 \quad (4)$$

ただし、 $\|\cdot\|_F$ は行列要素の2乗の和の平方根であり、行列のフロベニウスノルムと呼ばれる。この目的関数の重要な特徴として、式(4)はフロベニウスノルムの定義より、行列 $I - AM$ の各列の2-ノルムの2乗の和に変形できる点を持っている。

$$\begin{aligned} F(M) &= \|I - AM\|_F^2 \\ &= \sum_{j=1}^n \|e_j - Am_j\|_2^2 \end{aligned} \quad (5)$$

ただし、 e_j, m_j はそれぞれ行列 I, M の第 j 列を表している。式(5)を最小にすることは、次の各 f_j を最小にすることと同値である。

$$f_j(m_j) = \|e_j - Am_j\|_2^2 \quad j = 1, 2, \dots, n \quad (6)$$

この式から f_j は独立に計算できるので、並列計算機への実装に適している。

2.2 最小残差法

この節は、式(5)の各 $\|e_j - Am_j\|_2^2$ を最小にすることで構成する行列の列に関連したアルゴリズムについて考える。このアルゴリズムでは、次の n 個の線形問題を解くことにより、最小化を行うことになる。

$$Am_j = e_j, \quad j = 1, 2, \dots, n \quad (7)$$

これらの問題は、MR 法やリスタートをしない GMRES 法などのような非対称な問題を取り扱う降下法に基づくアルゴリズムを数ステップ実行することで解くことができる。最初の m_j は、初期値 M_0 の第 j 列であり、 M_0 は、次のように決める⁵⁾。

$$M_0 = \frac{1}{\|AA^T\|_1} A^T \quad (8)$$

2.2.1 パラメータ τ

近似逆行列 M を構成する目的はあくまで前処理をするためであり、厳密な近似逆行列を求める必要はない。そこで、任意の j に対応する式 (7) に適用する最小残差法の反復回数を τ で定義し、 $\tau = 1, 2, 3$ として比較検討を行う。

2.3 フィルインの考慮

本来の問題の係数行列は非常に疎な構造をしており、それを利用して実際の計算は行なわれる。従って、前処理行列を構成する場合でもその疎性を保持することは重要である。もしその点を考慮しなければ、極めて大きな計算時間を要することになるかもしれない。

前節で述べた MR 法をそのまま実行すれば、導き出される行列 M は係数行列 A に比べて疎性が大きく落ちており、反復法の実行に多くの計算時間を要するかもしれない。そこで、初期値 M_0 では要素がゼロであったが、反復を経て得られた前処理行列 M において非ゼロになってしまった位置、つまりフィルイン位置を考慮する必要がある。

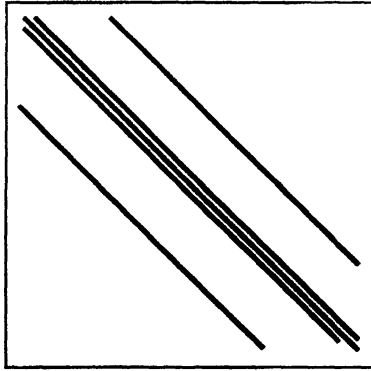
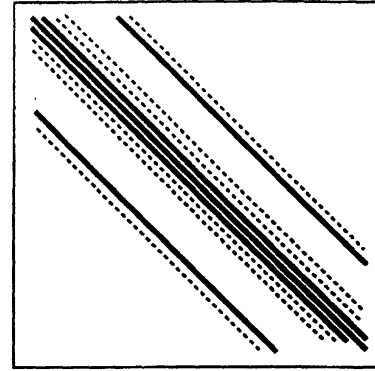
Chow ら²⁾ は棄却条件 *droptol* を定め、(i) フィルイン要素の値が *droptol* 以下であればその要素をゼロとする方法と、(ii) 列における非ゼロ要素の最大数 *lfil* を定める方法、の 2 つを提案している。今後、このような前処理を組み合わせた GMRES(m) 法を MR(*lfil*, *droptol*, τ)+GMRES(m) 法、または単に MR(*lfil*, *droptol*, τ) と記述する。次に、この 2 つのパラメータについて考えることにする。

2.3.1 パラメータ *lfil*

原メータ *lfil* は、近似逆行列 M の各列 m_j における非ゼロ要素の最大数である。現在、この値は理論的に決定できないため経験に頼るしかない。数値例では、係数行列は 5 重対角行列の構造を持っているので、*lfil* = 5 と *lfil* = 11 の 2 通りを考えることにする。⁶⁾

- *lfil* = 5 : 係数行列と同じ構造 (フィルインをしない)
- *lfil* = 11 : 対角列の本数を係数行列の約 2 倍に増やす (フィルインをする)

lfil = 11 の場合、元々存在する 5 本の対角列の近傍にフィルインを付加することにする。 M の構造をそれぞれ図 1 と図 2 に示す。

図 1 行列 M の構造 ($lfil = 5$ の場合)図 2 行列 M の構造 ($lfil = 11$ の場合)

パラメータ $lfil$ に対応する M の列ベクトル \mathbf{m}_j の非ゼロ要素の行インデックス集合を $I(lfil)$ で定義するすると、式 (7) は次のような式に書き換えられる。

$$A\mathbf{m}_j(I(lfil)) = \mathbf{e}_j, \quad j = 1, 2, \dots, n \quad (9)$$

$$A\tilde{\mathbf{m}}_j = \mathbf{e}_j, \quad j = 1, 2, \dots, n \quad (10)$$

ただし、式 (9) は、インデックス集合を省略して式 (10) のように記述する。

2.3.2 パラメータ $droptol$

$droptol$ はフィルインの閾値である。つまり、近似逆行列の要素 m_{ij} が $droptol$ 未満であれば 0 とする。drop 関数として数式で表現すると、次のようになる。

$$drop(m_{ij}) = \begin{cases} m_{ij} & (m_{ij} \geq droptol) \\ 0 & (m_{ij} < droptol) \end{cases} \quad (11)$$

2.4 ヘッセンベルグ行列の固有値と係数行列の固有値の関係

第 3 節の数値実験では、係数行列の固有値分布の変化について述べる。しかし、係数行列は大型であるため、その固有値を計算することは比較的難しい。その代わりに、GMRES(m) 法のアーノルディ過程で作られるヘッセンベルグ行列 H_m の固有値を計算する。本節では、ヘッセンベルグ行列の固有値 (係数行列の Ritz 値) と係数行列の固有値の関係について述べる。まず、アーノルディ原理の行列表現形として次式が成り立つ。

$$AV_m = V_m H_m + h_{m+1,m} \mathbf{v}_m \mathbf{e}_m^T \quad (12)$$

ただし、 θ_i : H_m の固有値 (Ritz 値)、 \mathbf{w}_i : θ_i に対応する H_m の固有ベクトルであり、 i は 1 以上 m 以下の整数である。つまり、

$$H_m \mathbf{w}_i = \theta_i \mathbf{w}_i \quad (13)$$

表 1 Origin 2400 の仕様

OS	IRIX 6.5.12
プロセッサ	MIPS R12000 300MHz × 16
メモリ	8GB
メモリバンド幅	680MB/秒 (実効) 780MB/秒 (ピーク)

式 (12) の右から w_i を掛けると

$$AV_m w_i = V_m H_m w_i + h_{m+1,m} v_m e_m^T w_i \quad (14)$$

$$AV_m w_i = \theta_i V_m w_i + h_{m+1,m} v_m \sigma_{m,i} \quad (15)$$

ただし, $\sigma_{m,i} = e_m^T w_i$ は w_i の第 m 要素である.

$$\|AV_m w_i - \theta_i V_m w_i\| = \|h_{m+1,m} v_m \sigma_{m,i}\| \quad (16)$$

$$= |h_{m+1,m} \sigma_{m,i}| \quad (17)$$

従って, θ_i を中心とする半径 $|h_{m+1,m} \sigma_{m,i}|$ の円内に行列 A の固有値が存在する. 数値実験では, 係数行列 A の固有値分布に代えて, ヘッセンベルグ行列の固有値 (係数行列の Ritz 値) 分布の変化を調べる.

3 数値実験

本節では, GMRES(m) 法, MR(lfl , $droptol$, τ)+GMRES(m) 法による数値実験を行い, 得られたデータを考察する. 数値実験は, 表 1 の仕様を持つ Silicon Graphics 社の分散共有メモリ型並列計算機 Origin2400 を使って行った. また, プログラム言語は C 言語を使用した. なお, プログラムを Origin2400 に実装するにあたり, pthread ライブラリを使って以下の部分を並列化した.

- ベクトルとベクトルの加算
- ベクトルのスカラー倍
- ベクトルの内積
- 行列とベクトルの乗算

行列およびベクトルの要素は, 各スレッドに均等に割り当てるものとする. 連立 1 次方程式の次元を n , CPU の数を p , 1 個の CPU に対して 1 個のスレッドを与えるとすると, 1 番目のスレッドは, ベクトルの第 1 行から第 \hat{n} 行の要素を担当する. ただし, $\hat{n} = n/p, (n \bmod p) = 0$

表 2 数値例 1 ($Dh = 2^{-4}$) : MR(lfl , $droptol$, τ) 法と GMRES(20) 法を組み合わせた算法の結果

算法	MR 法の計算時間 (秒)	総計算時間 (秒)	GMRES(20) 法の反復回数
GMRES(20)	0.00	31.91	1127
MR(5, $-\infty$, 1)	0.03	19.31	554
MR(5, $-\infty$, 2)	0.06	16.92	487
MR(5, $-\infty$, 3)	0.08	—	—
MR(5, 0.001, 1)	0.03	18.82	537
MR(5, 0.001, 2)	0.06	17.22	487
MR(5, 0.001, 3)	0.08	—	—
MR(5, 0.01, 1)	0.03	18.53	537
MR(5, 0.01, 2)	0.06	17.21	487
MR(5, 0.01, 3)	0.08	—	—
MR(5, 0.1, 1)	0.04	19.18	537
MR(5, 0.1, 2)	0.05	17.04	487
MR(5, 0.1, 3)	0.08	—	—
MR(11, $-\infty$, 1)	0.06	40.42	985
MR(11, $-\infty$, 2)	0.11	19.37	472
MR(11, $-\infty$, 3)	0.15	19.02	456
MR(11, 0.001, 1)	0.05	29.37	766
MR(11, 0.001, 2)	0.11	18.49	452
MR(11, 0.001, 3)	0.16	18.69	440
MR(11, 0.01, 1)	0.06	30.10	766
MR(11, 0.01, 2)	0.11	18.34	452
MR(11, 0.01, 3)	0.16	18.32	442
MR(11, 0.1, 1)	0.05	46.35	1201
MR(11, 0.1, 2)	0.11	20.58	526
MR(11, 0.1, 3)	0.16	17.34	436

“—” は、GMRES 法が収束しないため、結果が得られないことを示す

である。2 番目のスレッドは、第 $(\hat{n} + 1)$ 行から第 $2\hat{n}$ 行の要素を担当する。同様に、 i 番目のスレッドは、第 $\hat{n}(i - 1) + 1$ 行から第 $\hat{n}i$ 行の要素を担当する。各々のスレッドがベクトル v の要素を担当する様子を式 (18) に示す。行列の場合もベクトルの場合と同様に、各スレッドは上から順番に \hat{n} 行 n 列の長方形列を担当する。上記の演算で、ベクトルの内積以外は CPU 間で通信を行うことなく、計算を進めることが可能である。ベクトルの内積は、各スレッドが割り当てられた要素に関する局所的な内積を計算した後で、同期をとって最終的な内積を求める必要がある。

$$v^T = (\underbrace{v_1, \dots, v_{\hat{n}}}_{\text{thread 1}} \quad \underbrace{v_{\hat{n}+1}, \dots, v_{2\hat{n}}}_{\text{thread 2}} \quad \dots \quad \underbrace{v_{\hat{n}(i-1)+1}, \dots, v_{\hat{n}i}}_{\text{thread } i} \quad \dots) \quad (18)$$

前の 2.4 節で述べたように、数値実験では GMRES(m) 法のアーノルディ過程で構成されるヘッセンベルグ行列の固有値を計算する。その際、CLAPACK のルーチン “dhseqr.c” を用いる。これは実ヘッセンベルグ行列の固有値を求めるルーチンである。本稿の数値実験では Origin 2400 を使用し、その際 8 個のプロセッサを用いて並列計算を行う。反復回数の数え方

は、クリロフ部分空間の次元が1つ進む毎に反復を1回行ったものと数える。計算時間の計測は、各算法とも3回行なってその平均値を取る。数値実験で使用した収束条件などは、次のようにした。

- 収束判定条件： $\|r_k\|_2/\|r_0\|_2 \leq 1.0 \times 10^{-12}$
- 初期近似解： $x_0 = (0, 0, \dots, 0)^T$
- GMRES(m)法の最大反復回数：10000回
- 計算精度：倍精度
- スレッド数：8個

[数値例 1] 矩形領域 $\Omega = [0, 1] \times [0, 1]$ における2階の楕円型偏微分方程式のディリクレ境界値問題を考える。

$$\begin{aligned} -u_{xx}(x, y) - u_{yy}(x, y) + Du_x(x, y) &= G(x, y) \quad \text{on } \Omega = [0, 1]^2 \\ u(x, y) &= 1 + xy \quad \text{on } \partial\Omega \end{aligned}$$

ただし、メッシュ幅を $h = 1/129$ として5点中心差分近似で離散化し、得られる連立1次方程式は16384次元となる。右辺ベクトル b は、厳密解を $u(x, y) = 1 + xy$ と設定して決定する。さらに、係数行列の対角成分が1になるように対角スケーリングを行う。なお、 $Dh = 2^{-4}, 2^{-2}, 2^0, 2^2, 2^4$ の5通りで数値実験を行う。表中の“—”はGMRES(m)法が規定の最大反復回数内で収束条件を満たさなかったものである。

$Dh = 2^{-4}$ の場合の結果を表2に示す。総計算時間が最短だったのはMR(5, $-\infty$, 2)で、GMRES(20)法に比べて計算時間を約47%、反復回数を約57%短縮できた。 $lfil = 5$ で、 $\tau = 1$ もしくは $\tau = 2$ に設定したときは、前処理の効果が大きく出ている。 $lfil = 11$ にすると、前処理の効果が現れている算法もあるが、 $lfil = 5$ に比べると劣る場合が多い。特にMR(11, 0.1, 1)はGMRES(20)法に比べて計算時間が増大してしまった。このことから、特にフィルインをしなくても十分であることがわかる。図3と図4に計算時間及び反復回数に対する相対残差ノルムの収束の様子を示す。また、ヘッセンベルグ行列の固有値分布の様子を図7から図9に示す。いずれの場合も実数の固有値のみ存在している。前処理の効果が現れている場合は、最大固有値が大きく減少していることがわかる。

$Dh = 2^2$ の場合の結果を表3に示す。計算時間が最も短かったMR(11, 0.1, 3)はGMRES(20)法に対して、38%程度の計算時間の短縮に成功し、同時に反復回数を約56%短縮できている。 $Dh = 2^{-4}$ の場合とは異なり、近似逆行列を構成する上でフィルインをある程度行う必要のあることがわかる。それ以外のほとんどの算法は前処理の効果が現れず、近似解を得られない場合も増えてしまった。図5と図6に計算時間及び反復回数に対する相対残差ノルムの収束の様子を示す。先と同様に、ヘッセンベルグ行列の固有値分布の様子を図10から図12に示す。複素数の固有値も存在しているという点が $Dh = 2^{-4}$ の場合と異なって

表 3 数値例 1 ($Dh = 2^2$) : $MR(lfil, droptol, \tau)$ 法と GMRES(20) 法を組み合わせた算法の結果

算法	MR 法の計算時間 (秒)	総計算時間 (秒)	GMRES(20) 法の反復回数
GMRES(20)	0.00	19.70	678
MR(5, $-\infty$, 1)	0.03	23.95	679
MR(5, $-\infty$, 2)	0.06	23.33	663
MR(5, $-\infty$, 3)	0.08	17.03	481
MR(5, 0.001, 1)	0.03	21.74	620
MR(5, 0.001, 2)	0.06	-	-
MR(5, 0.001, 3)	0.08	-	-
MR(5, 0.01, 1)	0.03	21.78	617
MR(5, 0.01, 2)	0.05	-	-
MR(5, 0.01, 3)	0.08	-	-
MR(5, 0.1, 1)	0.03	21.74	626
MR(5, 0.1, 2)	0.05	15.57	435
MR(5, 0.1, 3)	0.07	-	-
MR(11, $-\infty$, 1)	0.06	31.59	760
MR(11, $-\infty$, 2)	0.11	19.85	473
MR(11, $-\infty$, 3)	0.16	14.77	357
MR(11, 0.001, 1)	0.06	23.72	581
MR(11, 0.001, 2)	0.11	19.26	478
MR(11, 0.001, 3)	0.15	16.78	406
MR(11, 0.01, 1)	0.06	23.89	581
MR(11, 0.01, 2)	0.11	19.25	478
MR(11, 0.01, 3)	0.16	16.79	406
MR(11, 0.1, 1)	0.06	23.72	591
MR(11, 0.1, 2)	0.11	18.56	455
MR(11, 0.1, 3)	0.16	12.21	299

“-” は、GMRES 法が収束しないため、結果が得られないことを示す

いる。前処理の効果が現れる場合は、固有値が 1 付近に密集するように分布が変化していることがわかる。

4 まとめ

本稿では、最小残差法による近似逆行列前処理と従来の GMRES(m) 法を組み合わせる算法を提案し、数値実験を行うことによりその有効性を示した。その際、3つのパラメータ $lfil$, $droptol$, τ のいくつか設定について考察した。

$MR(lfil, droptol, \tau) + GMRES(m)$ 法は、GMRES(m) 法が適用できる問題全てに適用することができ、問題に応じて3つのパラメータを適切に選択すれば、GMRES(m) 法の収束性を改善できる可能性がある。しかしながら、これらのパラメータの値を理論的に決定することは、現在のところ可能ではない。

パラメータ $lfil$ は近似逆行列の対角列の本数であるが、取り上げた数値例で係数行列と同じ本数でも前処理の効果が現れることがあることを示した。また、近似逆行列の構造が係数

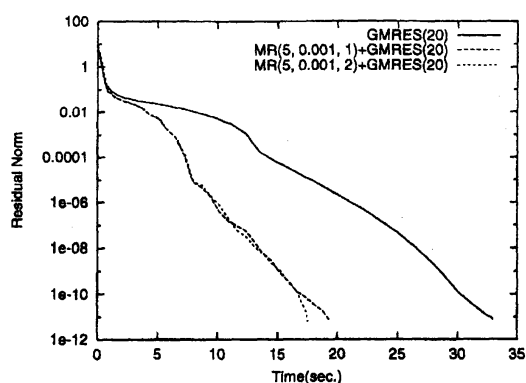


図 3 数値例 1($Dh = 2^{-4}$): 計算時間に対する相対残差

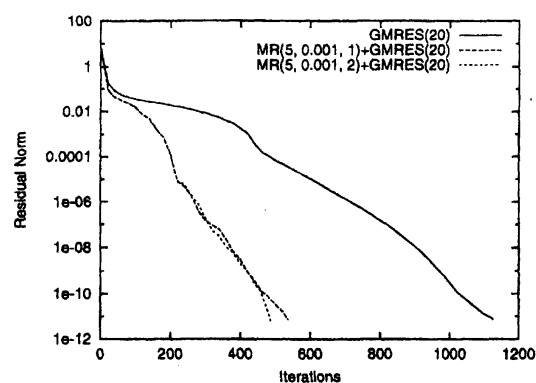


図 4 数値例 1($Dh = 2^{-4}$): 反復回数に対する相対残差

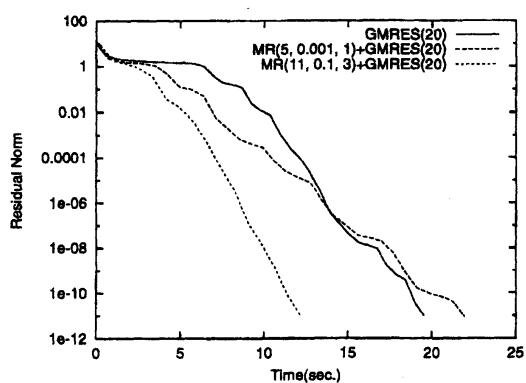


図 5 数値例 1($Dh = 2^2$): 計算時間に対する相対残差

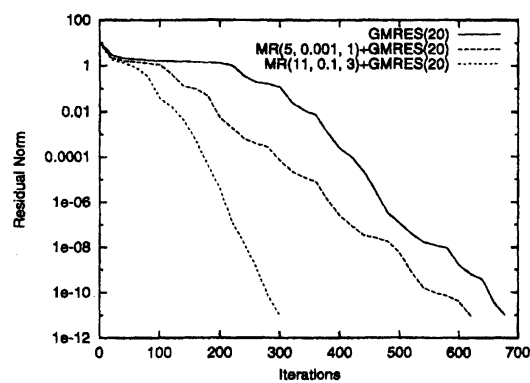


図 6 数値例 1($Dh = 2^2$): 反復回数に対する相対残差

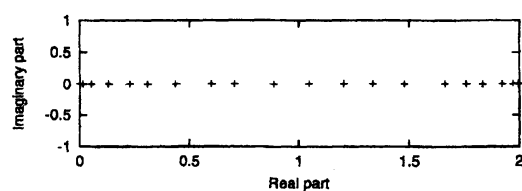


図 7 数値例 1($Dh = 2^{-4}$): GMRES(20) により生成される H_{20} の固有値分布

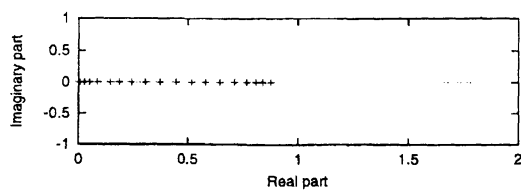


図 8 数値例 1($Dh = 2^{-4}$): MR(5, 0.001, 1)+GMRES(20) により生成される H_{20} の固有値分布

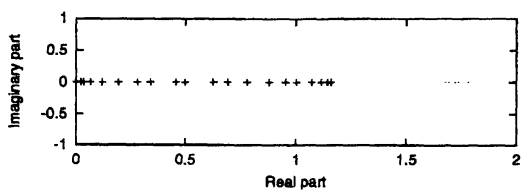


図 9 数値例 1($Dh = 2^{-4}$): MR(5, 0.001, 2)+GMRES(20) により生成される H_{20} の固有値分布

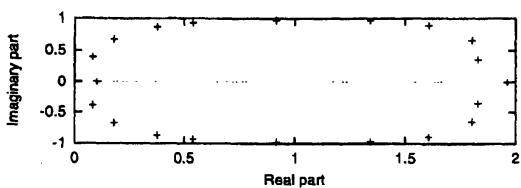


図 10 数値例 1($Dh = 2^2$): GMRES(20) により生成される H_{20} の固有値分布

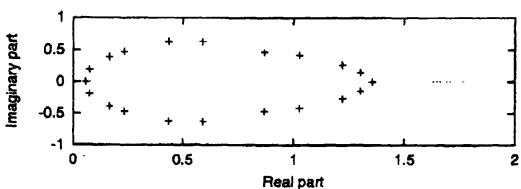


図 11 数値例 1($Dh = 2^2$): MR(5, 0.001, 1)+GMRES(20) により生成される H_{20} の固有値分布

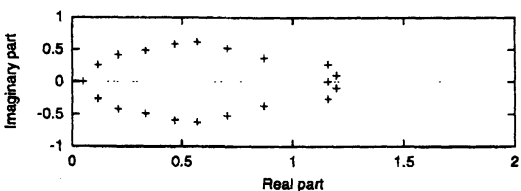


図 12 数値例 1($Dh = 2^2$): MR(11, 0.1, 3)+GMRES(20) により生成される H_{20} の固有値分布

行列と同じでは前処理の効果が現れない場合、対角列の本数を約2倍に増やすことで前処理の効果を得られる。しかし、一方で記憶容量が増大してしまうという問題も生じている。ここで、パラメータ $droptol$ は、近似逆行列の要素をふるいにかけ、小さい値の要素を捨てる役目を果たすものである。問題によっては、 $droptol$ を変化させると前処理の効果に大きな影響を与えることもある。次に、 τ はMR法を適用する回数を決定するパラメータである。これはMR法が反復法の一つであることから生れるものであり、数値実験によれば1回適用するだけでも十分である場合がある。また、3回適用しないと前処理の威力が発揮できないときもあることがわかった。

今後の課題は、前処理に関する3つのパラメータおよびGMRES(m)法のリスタート周期を問題に応じてある程度まで理論的に決定することである。リスタート周期を大きな値に設定すると、近似解を構成する上で必要な固有値、固有ベクトルを十分良い精度で得られるが、残念ながら記憶容量や1回の反復に必要な計算時間が増大する。一方、小さなリスタート周期の場合、記憶容量と1回の反復に必要な計算時間は減少するが、固有値や固有ベクトルの精度を落とすことになる。このようにリスタート周期は、算法の収束性に大きな影響を与える。また、 $lfil$, $droptol$, τ の関係も大きく影響してくる。以上のような課題が解決されると、最小残差法による近似逆行列による前処理はさらに有効な前処理の手法となる。

参考文献

- [1] Y. Saad and M. H. Schultz: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, pp. 856–869 (1986).
- [2] E. Chow and Y. Saad: Approximate inverse preconditioners via sparse-sparse iterations, *SIAM J. Sci. Comput.*, Vol. 19, No. 3, pp. 995–1023 (1998).
- [3] M. W. Benson: Iterative solution of large scale linear systems, Master's Thesis, Lakehead University, Thunder Bay, ON, Canada (1973).
- [4] M. W. Benson and P. O. Frederickson: Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems, *Utilitas Math.*, Vol. 22, pp. 127–140 (1982).
- [5] V. Pan and J. Reif: Efficient parallel solution of linear systems, in Proc. 17th Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, pp. 143–152 (1985).
- [6] T. Tanaka and T. Nodera: Effectiveness of approximate inverse preconditioning by using the MR algorithm on an Origin 2400, Civil-Comp Ltd, Stirling, Scotland, Proceedings of the Third International Conference on Engineering Computational Technology, Paper 44 (2002).